

CLAIMS

1 1. A system for performing remote procedure calls utilizing a markup language as a marshalling
2 format in which a client is passed a method name and associated parameters for a remote
3 procedure call, marshals said method name and parameters into a request markup language
4 format document and transmits said document, said system comprising:

5 a markup language remote procedure call server, said server receiving said method
6 name and associated parameters from said request markup language document;

7 said server invoking a method corresponding to said method name;

8 said server receiving return parameters from said invoked method;

9 said server generating a response markup language document containing said return
10 parameters;

11 a parser, said parser parsing said request markup language format document and
12 presenting said document to said server such that said server receives said method name and
13 associated parameters; and

14 wherein said request markup language document and said response markup language
15 document are encoded in a tokenized format.

1 2. A system for performing remote procedure calls utilizing a markup language as a marshalling
2 format in which a client is passed a method name and associated parameters for a remote procedure
3 call, marshals said method name and parameters into a request markup language format document
4 and transmits said document, as per claim 1, wherein said parser implements an event-based API.

3. A system for performing remote procedure calls utilizing a markup language as a marshalling format in which a client is passed a method name and associated parameters for a remote procedure call, marshals said method name and parameters into a request markup language format document and transmits said document, as per claim 2, wherein said parser translates the tokens of said tokenized request document into strings and presents said request document to said server as said strings.

4. A system for performing remote procedure calls utilizing a markup language as a marshalling format in which a client is passed a method name and associated parameters for a remote procedure call, marshals said method name and parameters into a request markup language format document and transmits said document, as per claim 3, wherein said parser translates the tokens into said strings using a code space generated offline.

5. A system for performing remote procedure calls utilizing a markup language as a marshalling format in which a client is passed a method name and associated parameters for a remote procedure call, marshals said method name and parameters into a request markup language format document and transmits said document, as per claim 2, wherein said parser presents said request document to said server as tokens.

6. A system for performing remote procedure calls utilizing a markup language as a marshalling format in which a client is passed a method name and associated parameters for a remote procedure

3 call, marshals said method name and parameters into a request markup language format document
4 and transmits said document, as per claim 1, wherein said parser implements a tree-based API.

1 7. A system for performing remote procedure calls utilizing a markup language as a marshalling
2 format in which a client is passed a method name and associated parameters for a remote procedure
3 call, marshals said method name and parameters into a request markup language format document
4 and transmits said document, as per claim 6, wherein said parser translates the tokens of said
5 tokenized request document into strings and presents said request document to said server as said
6 strings.

1 8. A system for performing remote procedure calls utilizing a markup language as a marshalling
2 format in which a client is passed a method name and associated parameters for a remote procedure
3 call, marshals said method name and parameters into a request markup language format document
4 and transmits said document, as per claim 7, wherein said parser translates the tokens into said
5 strings using a code space generated offline.

1 9. A system for performing remote procedure calls utilizing a markup language as a marshalling
2 format in which a client is passed a method name and associated parameters for a remote procedure
3 call, marshals said method name and parameters into a request markup language format document
4 and transmits said document, as per claim 6, wherein said parser presents said request document to
5 said server as tokens.

10. A system for performing remote procedure calls utilizing a markup language as a marshalling format in which a client is passed a method name and associated parameters for a remote procedure call, marshals said method name and parameters into a request markup language format document and transmits said document, as per claim 1, said system further comprising:

a servlet, said servlet receiving a request from said client to establish a connection with said server;

said servlet invoking said server upon receiving said request, said server establishing a connection with said client;

said server registering handler objects and associated methods which were to be invoked via said request markup language documents transmitted by said client;

wherein references to said handler objects and associated methods are stored in a hash table at said server.

11. A system for performing remote procedure calls utilizing a markup language as a marshalling format in which a client is passed a method name and associated parameters for a remote procedure call, marshals said method name and parameters into a request markup language format document and transmits said document, as per claim 10, wherein, prior to invoking said method, said server determines if said method corresponding to said method name is registered with said server via said hash table.

1 12. A system for performing remote procedure calls utilizing a markup language as a marshalling
2 format in which a client is passed a method name and associated parameters for a remote procedure
3 call, marshals said method name and parameters into a request markup language format document
4 and transmits said document, as per claim 1, wherein a transport protocol for transmitting said
5 request markup language document is HTTP.

1 13. A system for performing remote procedure calls utilizing a markup language as a marshalling
2 format in which a client is passed a method name and associated parameters for a remote procedure
3 call, marshals said method name and parameters into a request markup language format document
4 and transmits said document, as per claim 12, wherein said markup language is XML.

1 14. A system for performing remote procedure calls utilizing a markup language as a marshalling
2 format in which a client is passed a method name and associated parameters for a remote procedure
3 call, marshals said method name and parameters into a request markup language format document
4 and transmits said document, as per claim 12, wherein said request markup language document is
5 transmitted as the body of a HTTP-POST message.

1 15. A system for performing remote procedure calls utilizing a markup language as a marshalling
2 format in which a client is passed a method name and associated parameters for a remote procedure
3 call, marshals said method name and parameters into a request markup language format document
4 and transmits said document, as per claim 14, said system further comprising:

5 a servlet running as an extension to a HTTP service and receiving said HTTP-POST
6 message containing said request markup language document in the body of said HTTP-POST
7 message;

8 said servlet processing said HTTP-POST message, and

9 wherein upon determining the body of an HTTP-POST request is said request markup
10 language document, said servlet forwards said request markup language document to said server
11 upon which said parser parses said document and presents said document to said server such that
12 said server receives said method name and associated parameters.

16. A system for performing remote procedure calls utilizing a markup language as a marshalling
format in which a client is passed a method name and associated parameters for a remote procedure
call, marshals said method name and parameters into a request markup language format document
and transmits said document, as per claim 1, wherein said markup language is XML.

17. A system for performing remote procedure calls utilizing a markup language as a marshalling
format in which a server receives a request for a remote procedure call including a method name and
associated parameters in the form of a request markup language document, said server invokes a
method corresponding to said method name and transmits a response markup language document
containing return parameters returned from said invoked method, said system comprising:

a client, said client being passed said method name and associated parameters for said
remote procedure call;

8 said client generating said request markup language document including said method
9 name and associated parameters and sending said document to said server; and
10 wherein said request markup language document is encoded in a tokenized format.

1 18. A system for performing remote procedure calls utilizing a markup language as a marshalling
2 format in which a server receives a request for a remote procedure call including a method name and
3 associated parameters in the form of a request markup language document, said server invokes a
4 method corresponding to said method name and transmits a response markup language document
5 containing return parameters returned from said invoked method, as per claim 17, wherein a
6 transport protocol for transmitting said request markup language document is HTTP.

1 19. A system for performing remote procedure calls utilizing a markup language as a marshalling
2 format in which a server receives a request for a remote procedure call including a method name and
3 associated parameters in the form of a request markup language document, said server invokes a
4 method corresponding to said method name and transmits a response markup language document
5 containing return parameters returned from said invoked method, as per claim 18, wherein said
6 request markup language document is transmitted as the body of a HTTP-POST message.

1 20. A system for performing remote procedure calls utilizing a markup language as a marshalling
2 format in which a server receives a request for a remote procedure call including a method name and
3 associated parameters in the form of a request markup language document, said server invokes a

method corresponding to said method name and transmits a response markup language document containing return parameters returned from said invoked method, as per claim 17, wherein said markup language is XML.

21. A system for performing remote procedure calls utilizing a markup language as a marshalling format in which a server receives a request for a remote procedure call including a method name and associated parameters in the form of a request markup language document, said server invokes a method corresponding to said method name and transmits a response markup language document containing return parameters returned from said invoked method, as per claim 19, said system further comprising:

a parser, said parser receiving said response markup language document containing said return parameters;

said parser parsing said response markup language document and presenting said document to said client such that said client receives said return parameters; and

wherein said response markup language document is encoded in a tokenized format.

22. A system for performing remote procedure calls utilizing a markup language as a marshalling format in which a server receives a request for a remote procedure call including a method name and associated parameters in the form of a request markup language document, said server invokes a method corresponding to said method name and transmits a response markup language document containing return parameters returned from said invoked method, as per claim 21, wherein said

6 parser implements an event-based API.

1 23. A system for performing remote procedure calls utilizing a markup language as a marshalling
2 format in which a server receives a request for a remote procedure call including a method name and
3 associated parameters in the form of a request markup language document, said server invokes a
4 method corresponding to said method name and transmits a response markup language document
5 containing return parameters returned from said invoked method, as per claim 22, wherein said
6 parser translates the tokens of said tokenized response document into strings and presents said
7 response document to said client as said strings.

1 24. A system for performing remote procedure calls utilizing a markup language as a marshalling
2 format in which a server receives a request for a remote procedure call including a method name and
3 associated parameters in the form of a request markup language document, said server invokes a
4 method corresponding to said method name and transmits a response markup language document
5 containing return parameters returned from said invoked method, as per claim 23, wherein said
6 parser translates the tokens into said strings using a code space generated offline.

1 25. A system for performing remote procedure calls utilizing a markup language as a marshalling
2 format in which a server receives a request for a remote procedure call including a method name and
3 associated parameters in the form of a request markup language document, said server invokes a
4 method corresponding to said method name and transmits a response markup language document

5 containing return parameters returned from said invoked method, as per claim 22, wherein said
6 parser presents said response document to said client as tokens.

1 26. A system for performing remote procedure calls utilizing a markup language as a marshalling
2 format in which a server receives a request for a remote procedure call including a method name and
3 associated parameters in the form of a request markup language document, said server invokes a
4 method corresponding to said method name and transmits a response markup language document
5 containing return parameters returned from said invoked method, as per claim 21, wherein said
6 parser implements a tree-based API.

1 27. A system for performing remote procedure calls utilizing a markup language as a marshalling
2 format in which a server receives a request for a remote procedure call including a method name and
3 associated parameters in the form of a request markup language document, said server invokes a
4 method corresponding to said method name and transmits a response markup language document
5 containing return parameters returned from said invoked method, as per claim 26, wherein said
6 parser translates the tokens of said tokenized response document into strings and presents said
7 response document to said client as said strings.

1 28. A system for performing remote procedure calls utilizing a markup language as a marshalling
2 format in which a server receives a request for a remote procedure call including a method name and
3 associated parameters in the form of a request markup language document, said server invokes a

4 method corresponding to said method name and transmits a response markup language document
5 containing return parameters returned from said invoked method, as per claim 27, wherein said
6 parser translates the tokens into said strings using a code space generated offline.

1 29. A system for performing remote procedure calls utilizing a markup language as a marshalling
2 format in which a server receives a request for a remote procedure call including a method name and
3 associated parameters in the form of a request markup language document, said server invokes a
4 method corresponding to said method name and transmits a response markup language document
5 containing return parameters returned from said invoked method, as per claim 26, wherein said
6 parser presents said response document to said client as tokens.

1 30. A system for performing remote procedure calls utilizing a markup language as a marshalling
2 format in which a server receives a request for a remote procedure call including a method name and
3 associated parameters in the form of a request markup language document, said server invokes a
4 method corresponding to said method name and transmits a response markup language document
5 containing return parameters returned from said invoked method, as per claim 17, wherein said
6 method name and associated parameters are passed to said client via an *invoke* method of said client.

1 31. A system for performing efficient remote procedure calls, utilizing XML as a marshalling
2 format, where a method name and associated parameters are included in a remote procedure call
3 request XML document transmitted as the body of a HTTP-POST message, said system comprising:

4 a markup language remote procedure call server, said server receiving said method
5 name and associated parameters from said request XML document;

6 a servlet, said servlet receiving a request from said client to establish a connection
7 with said server;

8 said servlet invoking said server upon receiving said request, said server establishing
9 a connection with said client;

10 said server registering handler objects and associated methods which are able to be
11 invoked via said request XML documents transmitted by said client, references to said handler
12 objects and associated methods stored in a hash table at said server;

13 said server determining if a method corresponding to said method name is registered
14 utilizing said hash table and upon determining said method is registered, said server invoking said
15 method corresponding to said method name;

16 said server receiving return parameters from said invoked method;

17 said server generating a response XML document containing said return parameters;

18 a server side parser, said server side parser parsing said request XML format
19 document and presenting said document to said server such that said server receives said method
20 name and associated parameters; and

21 wherein said request XML document and said response XML document are encoded
22 in a tokenized format.

1 32. A system for performing efficient remote procedure calls, utilizing XML as a marshalling

format, where a method name and associated parameters are included in a remote procedure call request XML document transmitted as the body of a HTTP-POST message, as per claim 31, said system further comprising:

a client, said client being passed said method name and associated parameters for said remote procedure call;

said client generating said tokenized request XML document including said method name and associated parameters and sending said request document to said server;

a client side parser, said client side parser receiving said tokenized response XML document upon said server sending said document, and

said client side parser parsing said tokenized response XML document and presenting said document to said client such that said client receives said return parameters.

33. A system for performing remote procedure calls utilizing a markup language as a marshalling format in which a server receives a request for a remote procedure call including a method name and associated parameters in the form of a request markup language document, said server invokes a method corresponding to said method name and transmits a response markup language document containing return parameters returned from said invoked method, as per claim 31, wherein said server side parser implements an event-based API.

34. A system for performing remote procedure calls utilizing a markup language as a marshalling format in which a server receives a request for a remote procedure call including a method name and

3 associated parameters in the form of a request markup language document, said server invokes a
4 method corresponding to said method name and transmits a response markup language document
5 containing return parameters returned from said invoked method, as per claim 31, wherein said
6 server side parser implements a tree-based API.

1 35. A system for performing remote procedure calls utilizing a markup language as a marshalling
2 format in which a server receives a request for a remote procedure call including a method name and
3 associated parameters in the form of a request markup language document, said server invokes a
4 method corresponding to said method name and transmits a response markup language document
5 containing return parameters returned from said invoked method, as per claim 32, wherein said client
6 side parser implements an event-based API.

1 36. A system for performing remote procedure calls utilizing a markup language as a marshalling
2 format in which a server receives a request for a remote procedure call including a method name and
3 associated parameters in the form of a request markup language document, said server invokes a
4 method corresponding to said method name and transmits a response markup language document
5 containing return parameters returned from said invoked method, as per claim 32, wherein said client
6 side parser implements a tree-based API.

1 37. A method for implementing a remote procedure call in which a client generates a request
2 markup language document including a method name and associated parameters, said method

3 comprising:

4 receiving said request markup language document, said markup language document

5 encoded in a tokenized format;

6 parsing said document to determine said method name and associated parameters;

7 invoking a method corresponding to said method name and passing said parameters

8 to said method;

9 receiving return parameters from said method;

10 generating a response markup language document including said returned parameters,

11 said response markup language document encoded in a tokenized format.

1 38. A method for implementing a remote procedure call in which a client generates a request
2 markup language document including a method name and associated parameters, as per claim 37,
3 wherein said markup language is XML.

1 39. A method for implementing a remote procedure call in which a client generates a request
2 markup language document including a method name and associated parameters, as per claim 37,
3 said method further comprising:

4 encapsulating said tokenized response markup language document as the body in a
5 HTTP-POST message.

1 40. A method of performing remote procedure calls utilizing a markup language as a marshalling
2 format in which a server receives a request markup language document including a method name
3 and associated parameters, said server invokes a method corresponding to said method name and
4 returns a tokenized response markup language document including returned parameters returned
5 from said method, said method of performing remote procedure calls comprising:

6 receiving a method name and associated parameters;

7 generating a request markup language document including said method name and
8 associated parameters, said request markup language document encoded in a tokenized format;

9 sending said request markup language document to said server;

10 receiving said tokenized response markup language document returned from said
11 server; and

12 parsing said tokenized response markup language server to obtain said return
13 parameter, and

14 wherein said request markup language document and said response markup language
15 document are encoded in a tokenized format.

1 41. A method of performing remote procedure calls utilizing a markup language as a marshalling
2 format in which a server receives a request markup language document including a method name
3 and associated parameters, said server invokes a method corresponding to said method name and
4 returns a tokenized response markup language document including returned parameters returned
5 from said method, as per claim 40, wherein said markup language is XML.

1 42. A method of performing remote procedure calls utilizing a markup language as a marshalling
2 format in which a server receives a request markup language document including a method name
3 and associated parameters, said server invokes a method corresponding to said method name and
4 returns a tokenized response markup language document including returned parameters returned
5 from said method, as per claim 40, said method further comprising:

6 encapsulating said tokenized request markup language document as the body in a
7 HTTP-POST message.

10 43. An article of manufacture comprising a computer user medium having computer readable
11 code embodied therein implementing a remote procedure call in which a client generates a request
12 markup language document including a method name and associated parameters, said computer
13 readable code comprising:

14 computer readable program code receiving said request markup language document,
15 said markup language document encoded in a tokenized format;

16 computer readable program code parsing said document to determine said method
17 name and associated parameters;

18 computer readable program code invoking a method corresponding to said method
19 name and passing said parameters to said method;

20 computer readable program code receiving return parameters from said method;

21 computer readable program code generating a response markup language document
22 including said returned parameters, said response markup language document encoded in a tokenized

14 format.

1 44. An article of manufacture comprising a computer user medium having computer readable
2 code embodied therein performing remote procedure calls utilizing a markup language as a
3 marshalling format in which a server receives a request markup language document including a
4 method name and associated parameters, said server invokes a method corresponding to said method
5 name and returns a tokenized response markup language document including returned parameters
6 returned from said method, said computer readable code comprising:

7 computer readable program code receiving a method name and associated parameters;

8 computer readable program code generating a request markup language document

9 including said method name and associated parameters, said request markup language document
10 encoded in a tokenized format;

11 computer readable program code sending said request markup language document
12 to said server;

13 computer readable program code receiving said tokenized response markup language
14 document returned from said server; and

15 computer readable program code parsing said tokenized response markup language
16 server to obtain said return parameter, and

17 computer readable program code wherein said request markup language document
18 and said response markup language document are encoded in a tokenized format.